Extended Analysis of DES S-boxes

Lauren De Meyer, Begül Bilgin, and Bart Preneel

KU Leuven ESAT/COSIC and iMinds
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
firstname.lastname@esat.kuleuven.be

Abstract. For more than three decades, the Data Encryption Standard (DES) was one the most widely used cryptographic algorithms. It is still the dominating block cipher for banking applications. The DES was designed by IBM, verified by NSA and published by the National Bureau of Standards as a US Federal Information Processing Standard (FIPS) in 1977. The algorithm itself was fully public but the complete design criteria were only revealed by Coppersmith in 1994. He states that the IBM team was aware of differential cryptanalysis; the DES S-boxes are chosen to satisfy eight design criteria in order to resist this powerful attack. In their 1982 book, Meyer and Matyas state that the DES S-boxes were chosen so that they can be implemented with a minimum number of logic circuits. They mention that for an early design, in which not all of the design criteria are satisfied, the number of minterms varies between 40 and 48. However, for the final design the number of minterms is either 52 or 53, which is the smallest possible number that satisfies all the design criteria. Our research attempts to validate the IBM claims by generating a large number of candidate DES S-boxes satisfying specific criteria and by evaluating their number of minterms.
Keywords: DES, S-box, minterm, differential cryptanalysis

1 Introduction

The publication of the DES algorithm in 1977 by the US National Bureau of Standards was surrounded by controversy. First, there were complaints that the key length of 56 bits was too short to resist exhaustive key search attacks by well-funded opponents [4]. Second, while the algorithm details were published, only part of the design criteria were revealed, which lead to the speculation that DES contains a hidden security weakness or trapdoor. In 1993, Wiener showed an effective design for a US$ 1 million machine that can recover a DES key in 3.5 hours [6], which makes it plausible that the US government could recover DES keys by exhaustive search in the late 1970s. In 1989, Biham and Shamir [1] demonstrated an attack against DES in 1989 using a technique called differential cryptanalysis; in response IBM claimed that this attack was known to the designers of DES and that the design criteria for the DES S-boxes contributed to the defense against this technique. In 1994 Coppersmith, one of the designers of the DES, presented a list of eight design criteria for the S-boxes [2], claiming that these criteria were used for the creation of the eight original DES S-boxes. This seems to settle the question about the trapdoors. In spite of this, the role of the NSA in the design of DES and in particular its S-boxes is not fully clear, as the public statements from IBM and NSA on this matter seem to be not fully consistent.

In 1982 Meyer and Matyas, two other members of the DES design team at IBM, discuss the implementation of the S-boxes and more specifically the number of minterms necessary to implement them [5]. They claim that an early design produced S-boxes with a number of minterms between 40 and 48. As more design criteria were added, this distribution shifted to the range from 52 to 59. The left tail of the distribution was chosen in order to minimize

the size of the implementation of the DES in hardware. The goal of this paper is to verify this claim in the hope that this can throw some light on the generation of the DES S-boxes.

First we generated a large number of S-boxes that satisfy the design criteria of IBM published in 1994 [2]; this turned out to be more difficult than expected. We also generated weaker S-boxes. Next we compute the distribution of the number of minterms of all these S-boxes in order to validate the claim by Meyer and Matyas.

This paper is organized as follows. Section 2 discusses the design criteria of the DES S-boxes made public by Coppersmith and lists the distributions provided by Meyer and Matyas. In Section 3 our methods to find strong S-boxes are described, followed by a discussion of our results in Section 4; this discussion includes a comparison to the claims of Meyer and Matyas. Section 5 concludes the paper.

2 Properties of DES S-boxes

2.1 Design Criteria

In [2] the following design criteria for the DES S-boxes are listed:

(S-1) Each S-box has six input bits and four output bits.
(S-2) No output bit of an S-box should be too close to a linear function of the input bits. (That is, if we select any output bit position and any subset of the six input bit positions, the fraction of inputs for which this output bit equals the XOR of these input bits should not be close to 0 or 1, but rather should be near 1/2.)
(S-3) If we fix the leftmost and rightmost input bits of the S-box and vary the four middle bits, each possible 4-bit output is attained exactly once as the middle four input bits range over their 16 possibilities.
(S-4) If two inputs to an S-box differ in exactly one bit, the outputs must differ in at least two bits. (If $|\Delta I_{i,j}| = 1$, then $|\Delta O_{i,j}| \geq 2$, where $|x|$ is the number of 1-bits in the quantity x.)
(S-5) If two inputs to an S-box differ in the two middle bits exactly, the outputs must differ in at least two bits. (If $\Delta I_{i,j} = 001100$, then $|\Delta O_{i,j}| \geq 2$.)
(S-6) If two inputs to an S-box differ in their first two bits and are identical in their last two bits, the two outputs must not be the same. (If $\Delta I_{i,j} = 11xy00$, where x and y are arbitrary bits, then $\Delta O_{i,j} \neq 0$.)
(S-7) For any nonzero 6-bit difference between inputs, $\Delta I_{i,j}$, no more than eight of the 32 pairs of inputs exhibiting $\Delta I_{i,j}$ may result in the same output difference $\Delta O_{i,j}$.
(S-8) Similar to (S-7), but with stronger restrictions in the case $\Delta O_{i,j} = 0$, for the case of three active S-boxes on round i.

Each 6×4 S-box can be split into four 4×4 S-boxes (rows), where the leftmost and rightmost input bits of the large S-box are used to select one of the four smaller S-boxes. Therefore, we can make a distinction between criteria that are applicable to these smaller S-boxes and those that apply to the larger 6×4 S-boxes. Note that criterion (S-3) implies that each 4×4 S-box is a 4-bit permutation, which simplifies the search for S-boxes.

As mentioned earlier, the leftmost and rightmost input bits of a 6×4 S-box select one of the 4×4 S-boxes for which the four middle bits are the input. Since only the two middle input bits are varied in (S-5), this criterion can be completely verified for 4×4 S-boxes. If all 4×4 S-boxes that are used to generate a 6×4 S-box are verified, then the 6×4 S-box will always satisfy this criterion. Criteria (S-2) and (S-4) cannot completely validate a 4×4 S-box, but they can already be used to eliminate permutations, that will certainly lead to invalid S-boxes.

In order to find valid 6×4 S-boxes, we first create 4-bit permutations and validate them using criteria from (S-2) to (S-5). Next, these permutations can be combined to create S-boxes and tested with criteria (S-2), (S-4) and (S-6) to (S-8).

The following lemma shows that any S-box that satisfies the above criteria leads to three other solutions.

Lemma 1. *Let* $(0, 1, 2, 3)$ *be the rows of a valid S-box that satisfies all eight criteria. Then the S-boxes with the row orderings* $(1, 0, 3, 2)$*,* $(2, 3, 0, 1)$ *and* $(3, 2, 1, 0)$ *are also valid S-boxes.*

Proof. By criterion (S-4), if two inputs to an S-box differ in exactly one bit, the outputs must differ in at least two bits. Consider any column on row i. Then changing any one bit in the input will result either in a different column on the same row, or in the same column on row $i \oplus 1$ or $i \oplus 2$. It will never result in the same column on row $i \oplus 3$ since that would require an input difference of two bits. A rotation over all rows s.t. row $i' = i \oplus j$, where $j \in \{0, 1, 2, 3\}$ will still preserve this property. □

To make the Lemma 1 more clear, consider the first DES S-box S_1:

```
14 4 13 1 2 15 11 8 3 10 6 12 5 9 0 7
0 15 7 4 14 2 13 1 10 6 12 11 9 5 3 8
4 1 14 8 13 6 2 11 15 12 9 7 3 10 5 0
15 12 8 2 4 9 1 7 5 11 3 14 10 0 6 13
```

Observe that $S_1(0) = 14$ and $S_1(33) = S_1(\texttt{0b10001}) = 15$ which leads to a 2-bit input difference and 1-bit output difference. As we change the row ordering to (0,3,1,2), we get the following S-box S_1':

```
14 4 13 1 2 15 11 8 3 10 6 12 5 9 0 7
15 12 8 2 4 9 1 7 5 11 3 14 10 0 6 13
0 15 7 4 14 2 13 1 10 6 12 11 9 5 3 8
4 1 14 8 13 6 2 11 15 12 9 7 3 10 5 0
```

Here $S_1'(0) = 14$ and $S_1'(1) = 15$. Therefore, with this S-box a 1-bit input difference can lead to a 1-bit output difference which contradicts criterion (S-4). However, any other rotation mentioned in Lemma 1 leads to a valid S-box.

This means that for four valid 4-permutations, we do not have to verify all the 4! combinations but only $\frac{4!}{4} = 6$.

2.2 The Number of Minterms

The 6×4 S-box is a multi-output Boolean function, that can be expressed in canonical form using minterms. The number of minterms in the canonical form is the number of logical

ports necessary to implement the S-box. Matyas and Meyer claim that this number can be used as a heuristic for cryptographic strength, because the number of minterms needed for the final design of S-boxes far exceeded the amount needed for an earlier design. We repeat their main results below; for more details the reader is referred to [5]. Table 1 (left) shows the distribution of minterms after reduction for the early design of S-boxes, based on a sample of 18 S-boxes. It is left open which criteria were already included in this design, but the authors do admit that this design was nearly random.

Table 1: Distribution of minterms for the early variant (left) and the DES S-boxes (right)

# Minterms per S-box	# S-boxes
40	1
41	1
44	3
45	3
46	4
47	2
48	4

# Minterms per S-box	# S-boxes
52	3
53	7
54	9
55	22
56	16
57	20
58	4
59	2

As more design criteria were added, the number of minterms increased. The distribution for the final design of S-boxes is shown in Table 1 (right), based on a sample of 83 S-boxes. We assume that this design includes all criteria given by Coppersmith [2]. It is still unknown how many of these valid S-boxes exist. Matyas and Meyer used a sample of 83 S-boxes but the reason for this could simply be the limited computing capacity of the time.

According to Matyas and Meyer, the eight DES S-boxes were chosen from the left tail of this distribution (52 and 53 minterms) to make the implementation on a single chip as easy as possible.

In our research we have created a large amount of S-boxes using Coppersmith's criteria and we have compared the distribution of their minterms to those given by Meyer and Matyas. We are interested to know which criteria were part of the early design and whether generating S-boxes according to the Coppersmith's design criteria gives the minterm distribution of Table 1.

3 Finding Valid S-boxes

As described in Section 2.1, we focus our initial search on 4×4 S-boxes and then combine these to create valid 6×4 S-boxes. We first create random 4-bit permutations and evaluate them with criteria (S-2) to (S-5). Next these permutations were combined and tested with criteria (S-2), (S-4) and (S-6) to (S-8).

Our initial approach was to generate the 4×4 S-boxes randomly. However, as there are $16! \times \frac{4!}{4} = 125\,536\,739\,328\,000 \approx 2^{46.8}$ combinations of four permutations and only few valid

ones, random search turned out to be too slow to find a large number of valid permutations. Therefore, a more systematic method is necessary based on affine equivalence.

Definition 1 (Affine Equivalence). *Let A and B be $n \times n$ invertible linear mappings over GF(2) and a and b n-bit vectors over GF(2). Then, the invertible S-boxes S_1 and S_2 are affine equivalent if the affine equivalence relation $S_1(x) = B^{-1} \cdot S_2(A \cdot x \oplus a) \oplus b$ holds. The set of such affine equivalent S-boxes are called an affine equivalence class.*

In his PhD thesis, De Cannière [3, pp. 75–94] observes that for 4-bit permutations, 302 affine equivalence classes can be found. Since affine equivalence leads to the same algebraic and differential properties, it is more efficient to study one representative of each class. We hence tried to use permutations that are in the same affine equivalence classes as the permutations of the DES S-boxes. Table 2 shows to which affine equivalence classes the DES S-boxes belong with the numbering provided in [3].

Table 2: Equivalence classes of DES S-boxes – numbering from [3]

S-box	Equivalence class			
DES1	207	207	207	145
DES2	108	144	142	194
DES3	191	210	142	215
DES4	194	194	194	194
DES5	191	163	88	166
DES6	38	179	195	108
DES7	210	148	194	218
DES8	82	237	167	208

Since it is known that nonlinearity properties are unchanged by affine transformations, we can assume that all these 4-bit permutations satisfy criterion (S-2). The lists for each affine equivalence class were trimmed using criteria (S-4) and (S-5) and then the permutations were combined in the same way as in the corresponding DES S-box. For example the permutations from equivalence class 207 were used as first, second and third row in combination with a permutation from equivalence class 145. Finally, the invalid S-boxes were eliminated using criteria (S-2), (S-4) and (S-6) to (S-8). With this method, hundreds of valid S-boxes have been generated.

Finding valid S-boxes using the criteria in Coppersmith's paper was much more computationally intensive than expected; we only managed to create valid S-boxes using the equivalence classes of the DES S-boxes. This leads to the conclusion that it is likely that the designers of the DES algorithm had more efficient algorithms to create strong S-boxes; one can even speculate that there may have been additional criteria to simplify the search.

4 Discussion on the Number of Minterms

4.1 The final design

After generating 635 S-boxes using the affine equivalence classes of the DES permutations, we used the Espresso logic minimizer[1] to minimize their Boolean functions and calculate a distribution of the number of minterms necessary to implement them. Our distribution and that of Meyer and Matyas are shown in Figure 1; the hypothesis that the two distributions are the same is accepted with a significance of 9.15% by the Kolmogorov-Smirnov hypothesis test. However, when we calculate the number of minterms necessary to implement the 8 DES S-boxes, they appear to range from 52 to 55 and not from 52 to 53 as Matyas and Meyer suggested. The most plausible explanation for this small difference is that they have used a more powerful minimization tool.

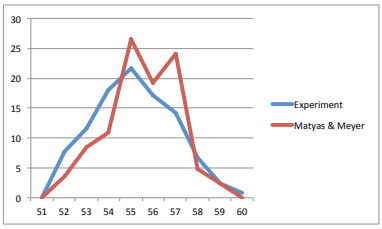
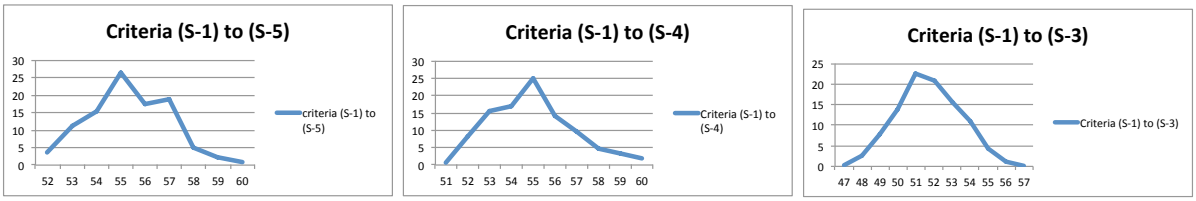


Fig. 1: Distribution of the number of minterms of S-boxes compared with the distribution of Meyer and Matyas

4.2 The Early Design

Meyer and Matyas also provide a distribution of the number of minterms for an early design of the S-boxes, situated between 40 and 48 minterms. To identify the criteria for this early design, we repeat our generation process for S-boxes that satisfy fewer criteria. Leaving out criteria from (S-5) to (S-7) does not cause a significant change. When we leave out (S-4), the minimum number of minterms becomes 47, which still does not correspond to the distribution of Meyer and Matyas.



(a) With criteria from (S-1) to (S-5) (b) With criteria from (S-1) to (S-5) (c) With criteria from (S-1) to (S-5)

Fig. 2: Distributions of the number of minterms of S-boxes for some of the criteria

[1] `http://en.wikipedia.org/wiki/Espresso_heuristic_logic_minimizer`

Only when the non-linearity criterion (S-2) is left out, a significant shift towards the left can be noticed, producing a distribution that corresponds to the Meyer and Matays distribution for the early design with a significance of 7.85%, again based on a Kolmogorov-Smirnov hypothesis test. However, the corresponding S-box is nearly random, since only criteria (S-1) and (S-3) are applied.

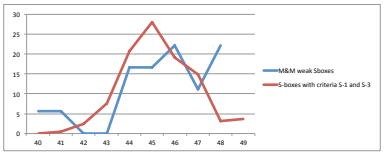


Fig. 3: Distribution of the number of minterms for S-boxes satisfying only criterion (S-1) and (S-3) compared to the distribution for the early design of Meyer and Matyas

5 Conclusion

We conclude that our search for S-boxes satisfying the eight criteria specified by Coppersmith [2] was much more computationally expensive than expected. This is rather surprising: according to Moore's law the computational power for the same cost has increased by a factor 25 million between 1975 and 2012; even if one takes into account the fact that the budget of IBM may have been two or three orders of magnitude larger than ours and IBM probably has spent much more time optimizing their code, this still leaves a large gap. Even with this gap, we had to resort to S-boxes that start from the existing S-boxes designed by IBM. Hence one can conclude that the algorithm used by IBM to generate DES S-boxes was much better than the best publicly known algorithm today. This leaves the open problem whether we can come up with better algorithms and whether we can find S-boxes that are not related to the DES S-boxes. The minimum number of minterms for our S-boxes was a little larger than for the DES S-boxes; however, it seems unlikely that this difference is the consequence of an unpublished design criterion.

The early design of the S-boxes described by Meyer and Matyas in 1982 [5] likely corresponds to a random combination of 4-bit permutations. One can wonder whether such a variant has seriously been considered by IBM, since by todays cryptographic standards such a DES variant is very weak.

References

1. E. Biham and A. Shamir. Differential cryptanalysis of DES-like cryptosystems. In *Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '90, pages 2–21. Springer-Verlag, 1991.
2. D. Coppersmith. The Data Encryption Standard (DES) and its strength against attacks. *IBM J. Res. Dev.*, 38(3):243–250, May 1994.
3. C. De Cannière. Analysis and design of symmetric encryption algorithms, PhD Thesis, KU Leuven, May 2007.
4. M. Hellman, R. Merkle, R. Schroeppel, L. Washington, W. Diffie, S. Pohlig, S., and P. Schweitzer. Results of an initial attempt to cryptanalyze the NBS Data Encryption Standard. *Information Systems Lab., Dept. of Electrical Eng., Stanford Univ.*, 1976.
5. C. H. Meyer and S. M. Matyas. Cryptography: A New Dimension in Computer Data Security. *John Wiley & Sons*, 1982.
6. M.J. Wiener. Efficient DES key search. *School of Computer Science, Carleton University*, 1993.